Journal of Automation and Control Engineering Vol. 3, No. 6, December 2015

# Control of Electric Motor Machine Tools Using Self-Organizing Fuzzy Logic Controller

Kyeong Mi Lee, Donggil Kim, and Dongik Lee
School of Electronics Engineering, Kyungpook National University, Daegu, South Korea
Email: rudal2419@naver.com, {eastroad, dilee}@ee.knu.ac.kr

*Abstract*—**For the control of electric motor which drives machine tools, it is important to achieve a high precision and accuracy. In general, PID-type controllers, which are widely used for industrial machine tools, exploit a Ziegler-Nichols based tuning method. However, it is difficult to perform real-time parameter tuning to obtain good performances under various operating modes. This paper addresses the application of a self-organizing fuzzy logic controller (SOFLC) to actively deal with the real-time variation of the plant's characteristics. SOFLC is a heuristic methodology where the control rules are generated and improved automatically according to the evaluation of the system's behavior, and to modify the control rule based on the system performance evaluation. The simulation is performed using Matlab/Simulink and the results of a comparison study between the proposed SOFLC controller and a PID type controller are presented to explore the performance and effectiveness of the proposed algorithm.**

*Index Terms*—**Self-Organizing fuzzy logic control, PID, machine tools, motor control**

## I. INTRODUCTION

Industrial machine tools are versatile manufacturing machines for making machine parts, where many performance criteria are needed to makes those machines work faster and with a high precision. Thus, it is important to control the electric motor that drives those machine tools under time-varying operating environment. A wide variety of control strategies have been proposed and discussed to be applied for precise motion control. However, in the design of control system for machine tools, it is important to take into account the significant level uncertainties caused by various materials with different characteristics. Therefore, the control system must have the capability of tolerating time-varying system parameters. In addition, as machine tools employ wide range of communication networks, the control system should be able to compensate randomly varying time delay in the measurements and actuations.

Currently the most commonly control strategy in industry is a PID-type controller thanks to its simple structure and easy implementation. In general, the PID control parameters can be manually determined by engineers using the trial-and-error method based on empirical knowledge or several model-based design methods. As described before, the resulting PID controller needs to adaptive in the presence of the change in operating environment. Typical auto-tuning methods for PID include Ziegler-Nichols, Astrom & Hagglund algorithm, and Cohen & Coonrule [4]. However, these approaches still suffer from difficulties in dealing with the rapid change of system characteristics [1]-[3] [5], so that the control system results in performance degradation.

This paper presents a self-organizing fuzzy logic controller (SOFLC) to actively address the time-varying plant characteristics including randomly varying network delays and actuator saturations. SOFLC is a heuristic methodology where the control rules are generated and improved automatically according to the evaluation of the system's behavior, and to modify the control action based on the system performance evaluation. The proposed SOFLC controller also contains a self-organizing level to modify the control rules by calculating the performance index. The simulation is performed using Matlab/Simulink and the results of a comparison study between the proposed SOFLC controller and a PID type controller are presented to explore the performance and effectiveness of the proposed algorithm.

This paper is organized as follows. Following this introduction, the self-organizing fuzzy logic controller is described in section 2. In section 3, the simulation results comparing the proposed SOFLC controller with a PID-type controller are presented. Finally, concluding remarks are followed in section 4.

## II. DESIGN OF SELF-ORGANIZING FUZZY LOGIC CONTROLLER

The SOFLC controller based on fuzzy logic is less sensitive to external environment or changes of the parameter on control object because it adapts to the plant through the change of control rules. In fuzzy logic control, the most difficult and significant problem is how to obtain the proper control rules for given plant. The control rules can be obtained from control engineering knowledge or the experience of the expert [6]. However, it takes significant time and cost to generate proper rules. To address this problem, the self-organizing fuzzy logic controller (SOFLC) was proposed by Procky and Mamdani [7].

The SOFLC is an adaptive controller with a mechanism for adjusting parameters. This approach is capable of generating and modifying the control rules by

---

Manuscript received August 21, 2014; revised December 24, 2014.

doi: 10.12720/joace.3.6.442-446

evaluating the system performance. That is, the SOFLC can be considered an advanced version of the fuzzy logic controller. The self-organizing fuzzy logic controller structure involves the self-organizing level on top of the conventional fuzzy logic controller consisting of the fuzzifier, rule base, fuzzy inference engine and defuzzifier, as shown in Fig. 1.

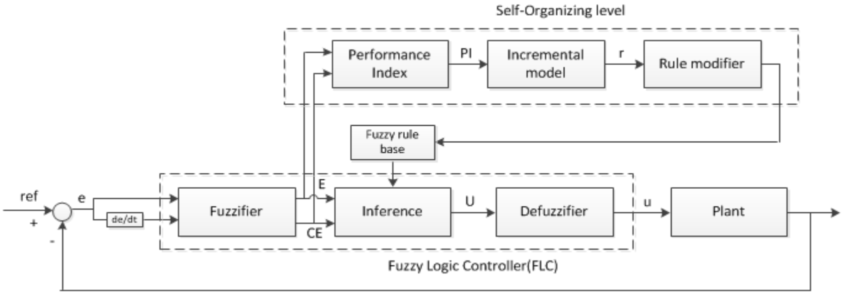


Figure 1. The structure of the self-organizing fuzzy logic controller [7].

The system performance is evaluated by using the change in error and the error between the reference input and actual system output. The performance index (PI) takes the form of a decision maker that issues the required output correction. An expert can determine the PI to achieve the desired performance based on the relation between the error and the change in error.

In this work, the performance table proposed by Yamazaki [8] is used as shown Table I. In Table I, linguistic logic represents numeral values ranging from -6 ~ +6. For example -6 means negative big and +6 means positive big, while the zero entries mean the states which require no correction. The incremental model is a constant in this work chosen to calculate the adjusting value to correct the control rules based on the performance evaluation.

TABLE I. PERFORMANCE TABLE

| | | Change in error | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Error | -6 | -6 | -6 | -6 | -6 | -6 | -6 | -6 | -5 | -4 | -3 | -2 | -1 | 0 |
| | -5 | -6 | -6 | -6 | -6 | -5 | -4 | -4 | -4 | -3 | -2 | -1 | 0 | 0 |
| | -4 | -6 | -6 | -6 | -5 | -4 | -3 | -3 | -3 | -2 | -1 | 0 | 0 | 1 |
| | -3 | -6 | -6 | -5 | -4 | -3 | -2 | -2 | -2 | -1 | 0 | 0 | 1 | 2 |
| | -2 | -6 | -5 | -4 | -3 | -2 | -1 | -1 | -1 | 0 | 0 | 1 | 2 | 3 |
| | -1 | -5 | -4 | -3 | -2 | -1 | -1 | 0 | 0 | 0 | 1 | 2 | 3 | 4 |
| | 0 | -5 | -4 | -3 | -2 | -1 | 0 | 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| | 1 | -3 | -2 | -1 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 3 | 4 | 5 |
| | 2 | -2 | -1 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 3 | 4 | 5 | 6 |
| | 3 | -1 | 0 | 0 | 0 | 1 | 2 | 2 | 2 | 3 | 4 | 5 | 6 | 6 |
| | 4 | 0 | 0 | 0 | 1 | 2 | 3 | 3 | 3 | 4 | 5 | 6 | 6 | 6 |
| | 5 | 0 | 0 | 1 | 2 | 3 | 4 | 4 | 4 | 5 | 6 | 6 | 6 | 6 |
| | 6 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |

The membership function for the error, the change of the error, and the controller output are shown in Fig. 2. The initial fuzzy control rules are shown in Table II. However, this initial control rules will change with the time-varying operating environment. Also max-min composition and center of gravity method are applied for inference and defuzzification.

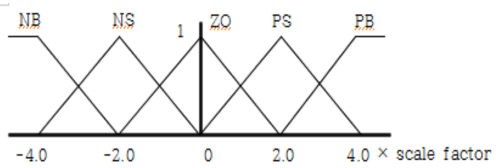


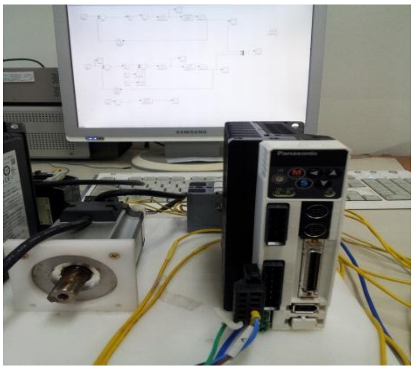
Figure 2. Membership functions of the error and the change of error.

TABLE II. FUZZY RULE BASE

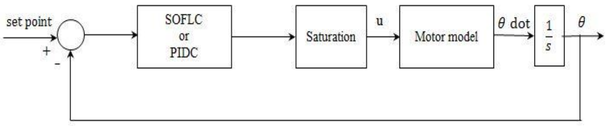
| ce<br>e | NB | NS | ZO | PS | PB |
|---|---|---|---|---|---|
| NB | NB | NB | NB | NS | ZO |
| NS | NB | NB | NS | ZO | PS |
| ZO | NB | NS | ZO | PS | PB |
| PS | NS | ZO | PS | PB | PB |
| PB | ZO | PS | PB | PB | PB |

## III. SIMULATIONS RESULTS

In this paper, the effectiveness of the proposed method is demonstrated using MATLAB/Simulink and compared with a PID-type controller. The experimental system and its block diagram considered in this work are depicted in Fig. 3. The experimental system consists of a TI-28335 based controller, a motor driver, and an electric motor, and an encoder to measure the angular position.



(a) Photo of experimental setup.



(b) Control system block diagram.

Figure 3. The configuration of experimental system.

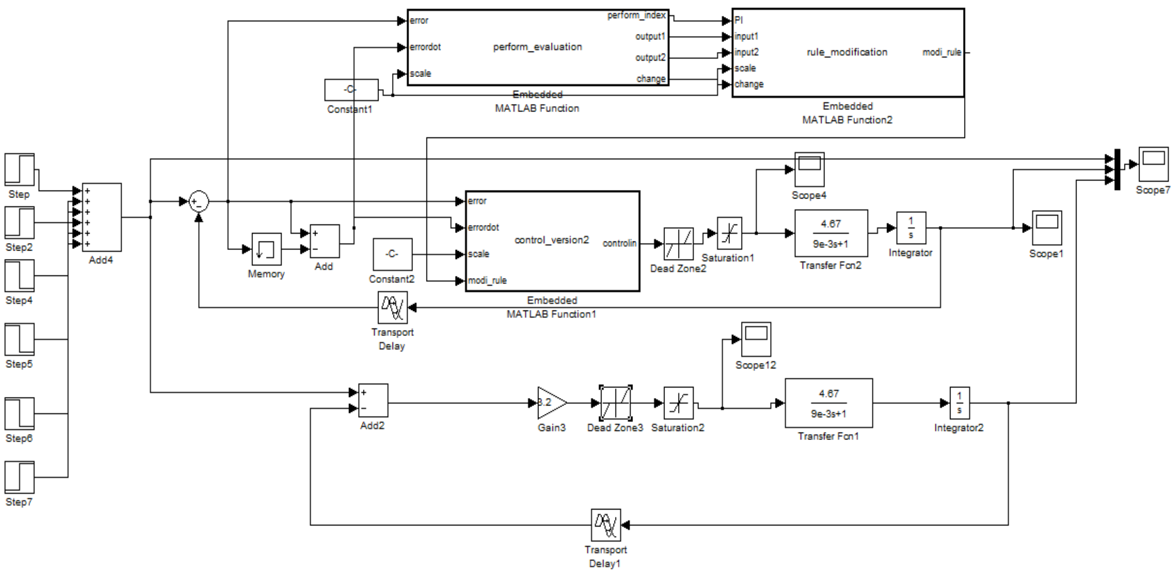


Figure 4. Simulink model with nonlinearities**.**

## A. *Modeling of the Experimental System*

The relationship between the input ($U$) to the motor driver and the angular velocity output ($w$) from the electric motor is derived with an experimental method, giving the following linear transfer function $G(s)$:

$$G(s) = \frac{\omega(s)}{U(s)} = \frac{4.67}{0.009s + 1} \tag{1}$$

To validate the accuracy of the model, the actual motor output and the simulated output using the model are compared in Fig. 5. In addition to the linear transfer function, additional network delays and nonlinearities such as driver saturation and dead zone are included in the Simulink model shown in Fig. 4.

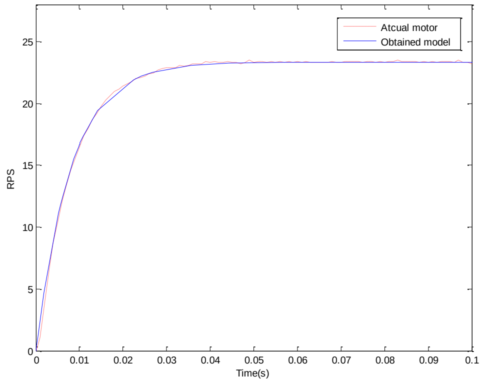


Figure 5. Comparison of motor velocity output between real and simulation.

## B. *Comparison of SOFLC with P Controller*

As the experimental system in this work is a type-1 system, the use of P (proportional) control for angular position seems reasonable. For a step response, the control requirements are given by 0.25s of settling time, 0.15s of rising time, and as little overshoot as possible. By considering these requirements, the P control parameter ($k_p$) was calculated to 3.2.

The simulation consists of two cases: (a) control under an ideal environment assuming that no delays or saturations are included; and (b) control under realistic environment involving the network induced delays and nonlinearities with the motor driver including saturation and dead zone.

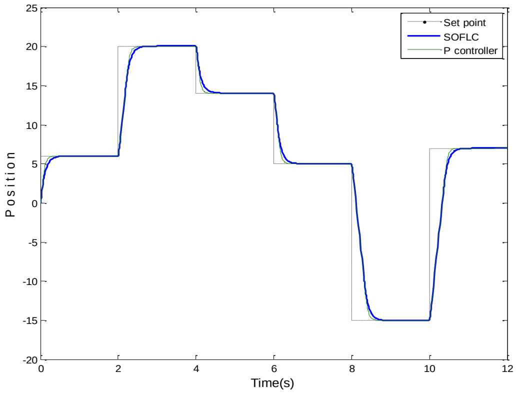


Figure 6. Comparison of SOFLC and P controller under ideal environment.

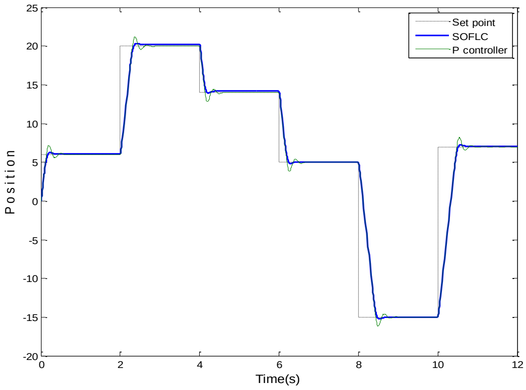


Figure 7. Comparison of SOFLC and P controller under realistic environment including network delay and nonlinearities

As shown in Fig. 6, in the ideal situation, both P control and SOFLC control satisfy the control requirements very well. The P controller shows even better performance in terms of rising time.

However, under the undesirable characteristics of time delays, saturations, and dead zones, which are varying with time, the P controller cannot satisfy the requirement in terms of overshoot. In contrast, although the SOFLC shows some steady state error in the early stage, the output satisfies the all requirements after 4s, thanks to the real-time adaptation of the control rules.

## IV. EXPERIMENT

In this paper, we also conduct an experiment as well as simulation to validate the effectiveness of the proposed method. The structure for experimental system is depicted in Fig. 8.

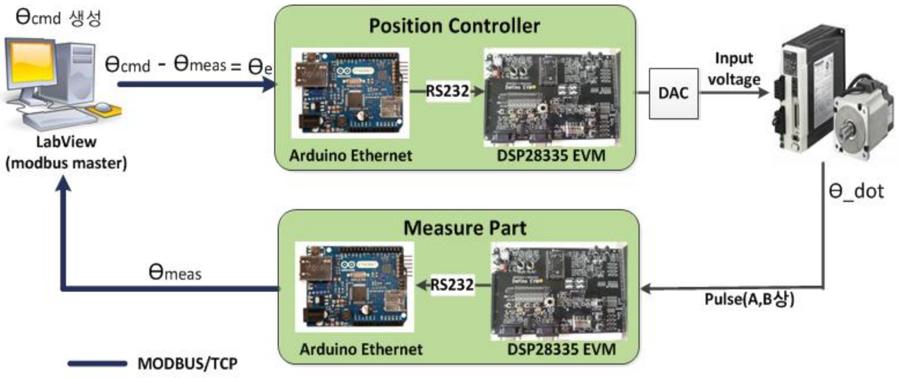


Figure 8. The structure of experimental system

The experimental system consists of Labview, arduinoethernet board, TI-28335 based controller, DAC(Digital Analog Converter), a motor drive, an electric motor, and an encoder. Labview conducts some work such as data logging, generating command value for position and transmitting error value. In controller, it receives the error value and performs self-organizing fuzzy logic algorithm or PID-type algorithm. In measure part, it calculates present position by counting pules of encoder with the electric motor and then transmits the position value to Labview. We used MODBUS/TCP [9] which is one of industry fieldbus to exchange data between Labview and controller, Labview and measure part. To communicate MODBUS/TCP, arduinoethernet board is used as MODBUS/TCP slave.

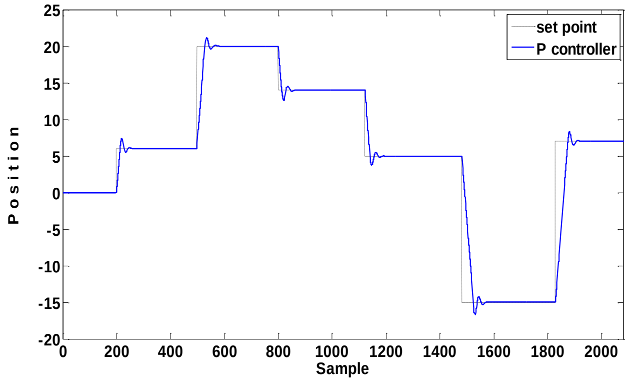


Figure 9. The experiment result when applying P controller

The experiment includes characteristic of time delay which generates by using network, non-linearies such as dead zone and saturation. The results of the experimental system are depicted in Fig. 9 and Fig. 10. It appeared similarly with the results of simulation. As shown Fig. 9, the P controller cannot satisfy the requirement in term of overshoot. On the other hands, when applying self-organizing fuzzy logic algorithm, the output satisfies the requirements because of adaptation the non-linearies and time delay by modifying the control rules in real-time.

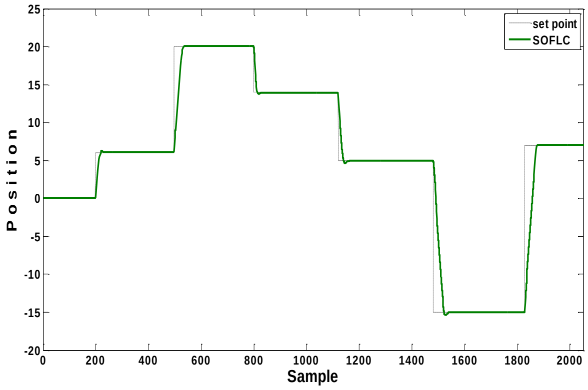


Figure 10. The experiment result when applying SOFLC

## V. CONCLUSION

This paper presented an adaptive controller algorithm based on self-organizing fuzzy logic controller which modifies the control rules by evaluating the performance of the plant. The proposed method has been compared with a P controller on an electric motor which are used for machine tools. The simulation results show that SOFLC is capable of dealing with time-varying undesirable characteristics of the plant by revising its initial control rules. And the experiment results also show that SOFLC can be applied with undesirable characteristics of the plant.

## ACKNOWLEDGMENT

This research was supported by the 2014 Doosan Infracore Co., Ltd. support program, Korea (201400640000).

## REFERENCES

[1] J. G. Ziegler and N. B. Nichols, "Optimum setting for automatic controllers," *Journal of Dynamic System, Measurement and control,* vol. 115, no. 2B, pp. 220-222, June 1993.

[2] D. Kim, D. Son, and D. Jeon, "Feed-system autotuning of a CNC machining center: Rapid system identification and fine gain tuning based on optimal search," *Precision Engineering,* vol. 36, no. 2, pp. 339-348, April 2012.

[3] W. S. Huang, C. W. Liu, and P. L. Hsu, "Precision control and compensation of servomotors and machine tools via the disturbance observer," *IEEE Transaction on Industrial Electronics,* vol. 57, no. 1, pp. 420-429, Jan. 2010.

[4] C. C. Hang, K. J. Astrom, and Q. G. Wang, "Relay feedback auto-tuning of process controllers–a tutorial review," *Journal of Process Control,* vol. 12, no. 1, pp. 143-162, Jan. 2002.

[5] P. Panda, C.C. Yu, and H. P. Huang, "PID tuning rules for SOPDT systems: Review and some new result," *ISA Transactions,* vol. 43, no. 2, pp. 283-295, April 2004.

[6] C. L. Chen and Y. M. Chen, "Self-organizing fuzzy logic controller design," *Computers in Industry,* vol. 22 , no. 3, pp. 249-267, October, 1993.

[7] T. J. Procyk and E. H. Mamdani, "A linguistic self-organising process controller," *Automatic,* vol. 15, pp. 15-30, 1979.

[8] T. Yamazaki and E. H. Mamdani, "On the performance of a rule-based self-organizing controller," in *Proc. IEEE Conf. Applications of Adaptive and Multivariable Control,* Hull, 1982, pp. 121-132.

[9] Modbus organization. MODBUS APPLICATION PROTOCOL SPECIFICATION V1.13. [Online]. Available: www.modbus.org/docs/Modbus_Application_Protocol_V1_13b.pdf

**Kyeong Mi Lee** was born in Seoul, Korea in 1990 and received B.S degree in Electronics Engineering from Kyungpook National University, Daegu, Korea, in 2013. She is presently a Master's Student of Electronics Engineering at Kyungpook National University, Daegu, Korea. Her research interests focuses on motor control of a machine tool and industrial networkedcontrol.

**Donggil Kim**received B.S degree in Electronics Computer Engineering and M.S degree in Electronics Engineering from Kyungpook National University, Daegu, Korea, in 2006 and 2008, respectively. He is presently a PhD's Student of Electronics Engineering at Kyungpook National University, Daegu, Korea. His research focuses on networked control, fieldbus and fault detection control.

**Dongik Lee** received his B.S degree and M.S. degree in Control Engineering from Kyungpook National University, Daegu, Korea, in 1987 and 1990, respectively. In 2002, he received a Ph D from Sheffield University, England. He is now an Assistant Professor of the school of Electronics Engineering, KyungpookNationl University. His research interest focuses on the design of real-time networked control for various safety-critical applications, including submarines, autonomous underwater vehicles and intelligent automobiles.